A FE Chip Software Emulator

Nikola Whallon

University of Washington

February 13th, 2017, US ITK Pixel Meeting

Introduction

ITk

- after Run 3, ATLAS inner tracker couldn't last in HL LHC radiation
- need new inner tracker which is radiation hard
- will use all-silicon design (SCT, Pixel, no TRT) and new radiation hard front-end chips

RD53A

- development chip to guide the ITk TDR
- contains 3 variants - all are radiation hard and meet the HL LHC ITk spec
- must be benchmarked in real life to make sure it meets spec

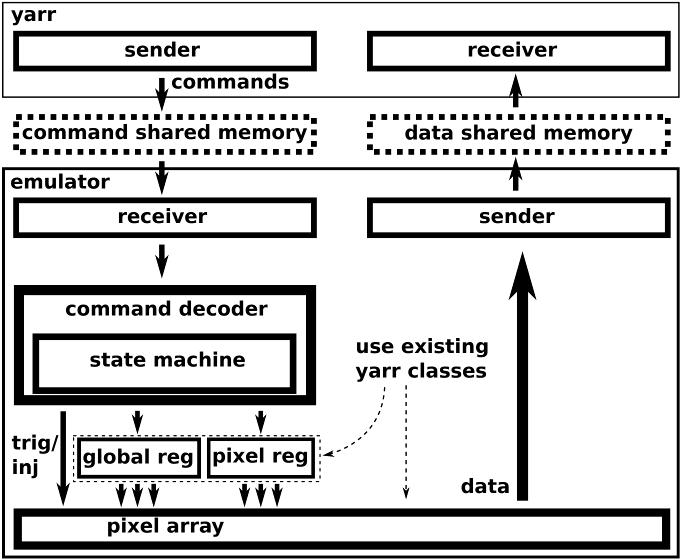
YARR - Yet Another Rapid Readout

- the DAQ and scan software being developed for RD53A at LBNL
- currently works with FEI4 (the front-end used for IBL) and FE65

Introduction to the Software Emulator

- Since the RD53A chip will not be available until mid-2017, I am writing a software emulator for the chip.
- The purpose of the emulator is to help development of the YARR DAQ software for the RD53A.
- YARR currently also supports FEI4 so that development with some hardware can occur.
- Along the same lines, the software emulator I am developing will support both FEI4 and RD53A - this allows us to fully cross check the system with FEI4.
- The FEI4 emulator is more or less complete, so I will show some results with using it today.

Emulator Flow Chart



This entire diagram has been implemented.

Modeling

Each of the n pixels have different threshold and noise behavior, dictated by the following:

$$\mathrm{thr}_n = v_n \cdot \mathrm{VthinFine} + v_n \cdot \mathrm{VthinCoarse} \cdot 128 - 30 \cdot \mathrm{TDAC}_n \quad (1)$$

$$\mathrm{noise} = \mathrm{Gauss}(0, n_n) \quad (2)$$

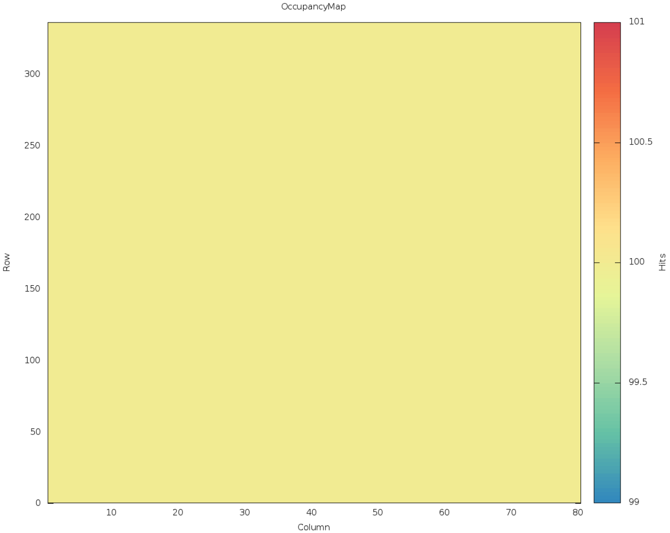
v_n and n_n are initialized once, and are initialized to:

$$v_n = \mathrm{Gauss}(22, 1) \quad (3)$$

$$n_n = \mathrm{Gauss}(150, 15) \quad (4)$$

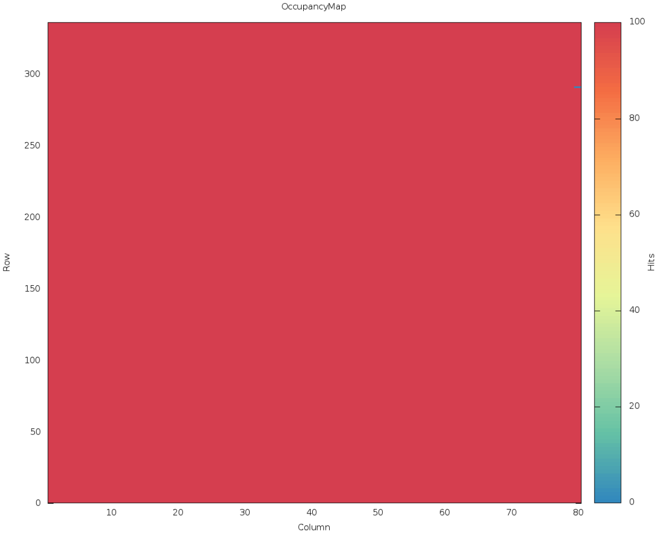
The step size for TDAC is fixed at 30 right now, but will change eventually to use TDACVbp.

Digital Scan



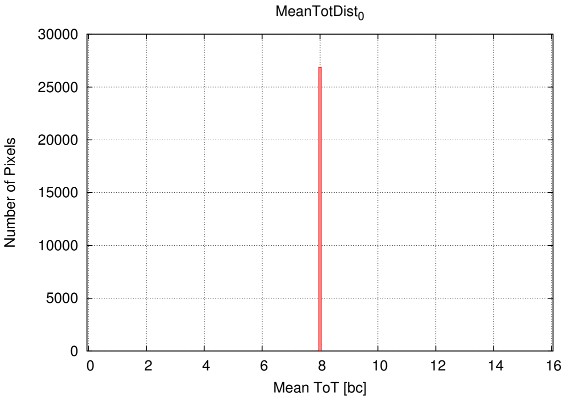
Digital scans look great.

Analog Scan



Analog scans look great.

ToT Scan

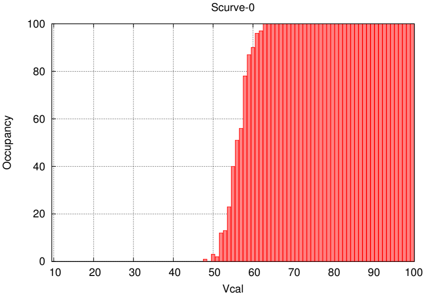


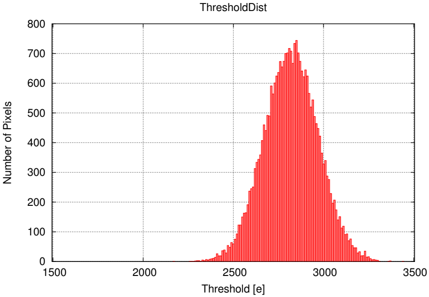
ToT is calculated using:

$$(\mathrm{inj} + \mathrm{noise} - \mathrm{thr}) \cdot \frac{9}{16000} + 1 \tag{5}$$

With current model parameters, it seems fixed. I can/have forced a wide(r) distribution by playing with the model parameters.

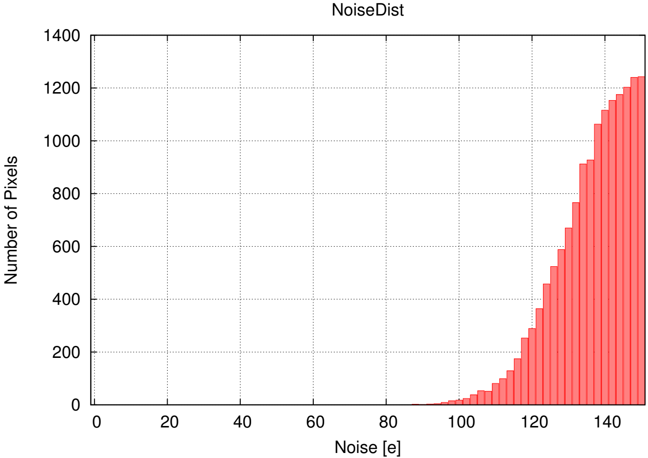
Threshold Scan





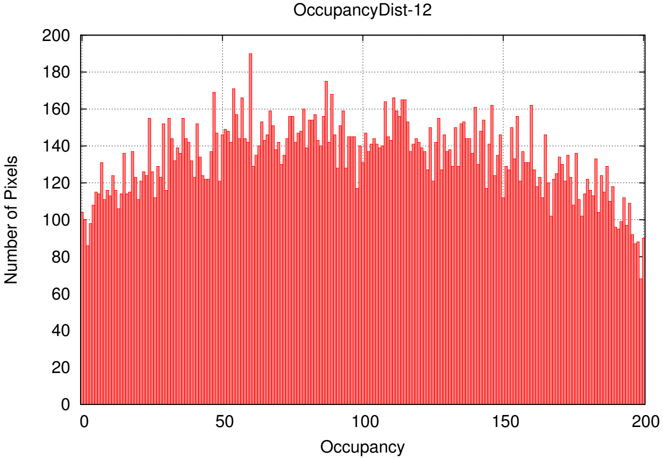
Threshold scans look great.

Threshold Scan Noise



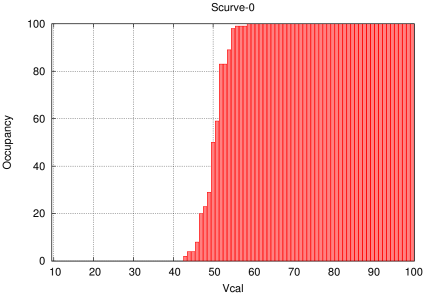
The noise looks spot on 150, where it should be.

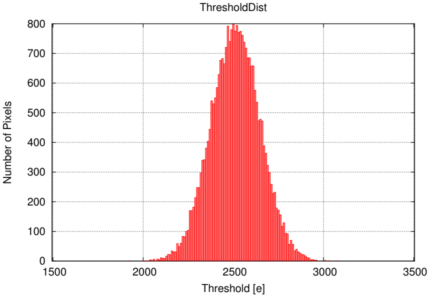
Global Threshold Tune



Global threshold tunes look great.

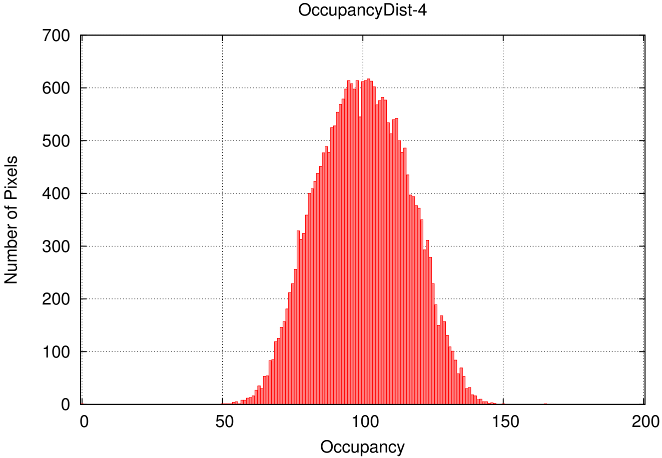
Threshold Scan After Global Threshold Tune





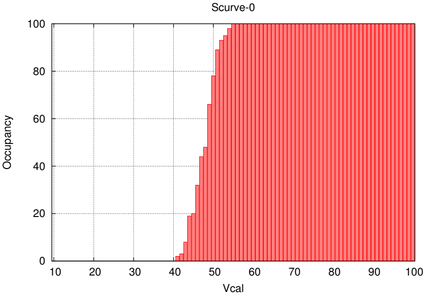
The global threshold scan seems to really work.

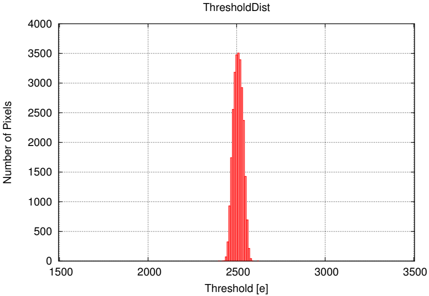
Pixel Threshold Tune



This looks great.

Threshold Scan After Pixel Threshold Tune





The pixel threshold tune seems to really work.

Summary and Next Steps

The FEI4 emulator seems to be functioning very well.

Documentation on how to use the emulator can be found here:
`http://yarr.readthedocs.io/en/devel/emulator.html`

Next steps involve refactoring a bit to make the emulator easy to accomodate any number of new/different FE chips (notably RD53A).